EUROPEAN
SPALLATION
SOURCE

Document Type	Engineering Manual
Document number	ESS-0508492
Date	February 11, 2020
Revision	1
State	Released
Confidentiality Level	Internal
Page	1 (16)

ICS Engineering Manual

for μTCA backplane clock distribution

	Name	**Role/Title**
Owner	Javier Cereijo Garcia	Timing System Engineer
Author	Javier Cereijo Garcia Simone Farina Jeong Han Lee	Timing System Engineer Embedded Systems Engineer Senior Controls Engineer

Contents

1 Overview

At European Spallation Source (ESS), Integrated Control System (ICS) uses the MicroResearch Finland (MRF) Timing System[1] as its timing system of the ESS site. The consistent and up-to-date engineering manual is essential for the ESS timing system.

1.1 Scope

- This document explains how to configure a μTCA system to send the event clock from a MTCA-EVR-300(U) to an Advanced Mezzanine Card (AMC) using the backplane.
- This document provides the configuration for the MTCA Carrier Hub (MCH) and EVR, and guidelines for best practices on the use of the backplane clocks in the AMCs.

1.2 Target audience

This document is targeted to ICS engineers and technical stakeholders of the ESS timing system. It is assumed that the target audience has a technical background in the MRF Timing System, the Experimental Physics and Industrial Control System (EPICS) development, and a Linux environment.

2 System description

MRF Technical Reference [see 1, p45] describes Event Receivers (EVRs) as:

> *Event Receivers decode timing events and signals from an optical event stream transmitted by an Event Generator. Events and signals are received at predefined rate the event clock that is usually divided down from an accelerators main RF reference. The event receivers lock to the phase event clock of the Event Generator and are thus phase locked to the RF reference. Event Receivers convert event codes transmitted by an Event Generator to hardware outputs. They can also generate software interrupts and store the event codes with globally distributed timestamps into FIFO memory to be read by a CPU.*

ICS uses and will use the following different types of EVRs:

- MTCA-EVR-300(U)
- PCIe-EVR-300DC

[1] http://www.mrf.fi/

One of the appealings of the MTCA-EVR-300(U) is its capability to send the event clock which is derived from and phase-locked to the main radio-frequency (RF) frequency to the AMCs via the backplane. For this the system should be configured in the appropriate way described in this document.

2.1 MTCA.4 standard

The microTCA standard, sometimes referred to as μTCA, uTCA or MTCA is short for Micro Telecommunications Computing Architecture. The bare bone components needed in every MTCA.4 system are a crate/chassis that includes a backplane and cooling units (CUs), a power supply and an MCH. The crate can have up to four slots for power modules (PMs), up to two for MCHs and up to 12 AMC slots. The crate itself is quite passive - in order to be able to use the the system at least one MCH has to be present in the system.

2.1.1 MCH

The MCH is inserted into a dedicated slot in the crate and manages the system. It takes care of routing connections between the system components and monitors the overall system status. It is also able to protect the system in case anything goes wrong.

Most of the configuration to send the event clock to an AMC is done in the MCH, since it's in charge of routing the connections over the μTCA backplane, including the clock lines.

2.2 MTCA-EVR-300(U)

Figure 1 shows the rough physical dimensions 181 × 148 mm^2 of the MTCA-EVR-300 card.

The MTCA-EVR-300 has a small form-factor pluggable (SFP) transceiver as an input from the Event Master (EVM) and several outputs: 4 front panel outputs, 16 front universal outputs (through the IFB-300 extension board), 8 backplane trigger lines and 2 backplane clock lines. The 16 front universal outputs are implemented through a micro-SCSI type connector for an interface board IFB-300, which has eight Universal I/O (Input/Output) slots. The MTCA-EVR-300U is identical but replaces the micro-SCSI connector for two Universal I/O slots.

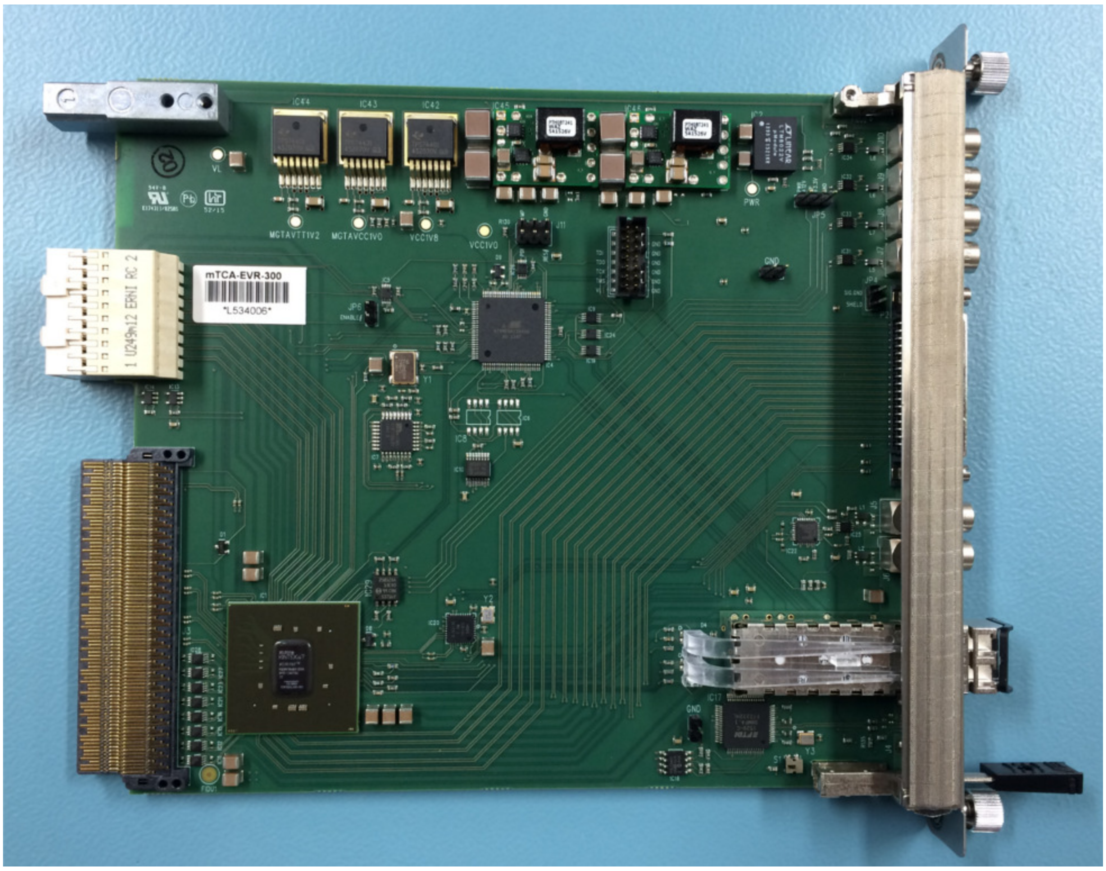


Figure 1 MRF MTCA-EVR-300 board.

3 System environment

Before describing the engineering procedure for sending the event clock through the backplane, it is mandatory to have a proper system environment that consists of specific hardware and software. The information shown in this chapter is used in the ICS lab at ESS.

3.1 Hardware

Table 1 shows the hardware list and its environment. It is possible to use different form factors than what is shown in the examples; for more information check the specific engineering manual.

Hardware	Info
MRF mTCA-EVM-300	
MRF mTCA-EVR-300(U)	
μTCA crate	Incl. PM, MCH
Concurrent Technologies AMC CPU	
Optical cables	LC, optical 850 nm

Table 1 Hardware list and its environment.

Figure 2 shows the MTCA-EVR-300U setup in the lab. From left to right, the power supply, MCH, central processing unit (CPU) and MTCA-EVR-300U.

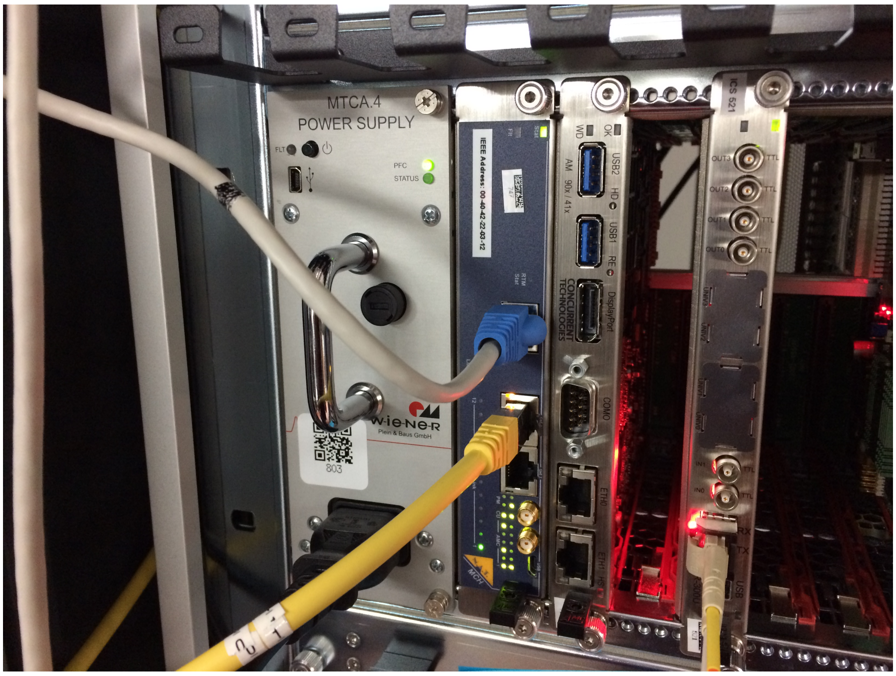


Figure 2 Hardware set up in the ICS lab.

3.2 Software

Table 2 shows the software list and its environment.

Item	Version Info.
CentOS Linux	`7.7.1908`
Kernel	`3.10.0-1062.9.1.el7.x86_64`
mrf kernel module	Version: `1` / srcversion `E3290AD048B5B57D2EAA55E`
EPICS base	`7.0.3.1`
e3-req	`3.1.2`
mrfioc2	E3 module ver. `2.2.0-rc7`
devLib2	E3 module ver. `2.9.0`

Table 2 Software and its version information.

4 Engineering procedure

This chapter explains how to configure the system to send the event clock via the backplane.

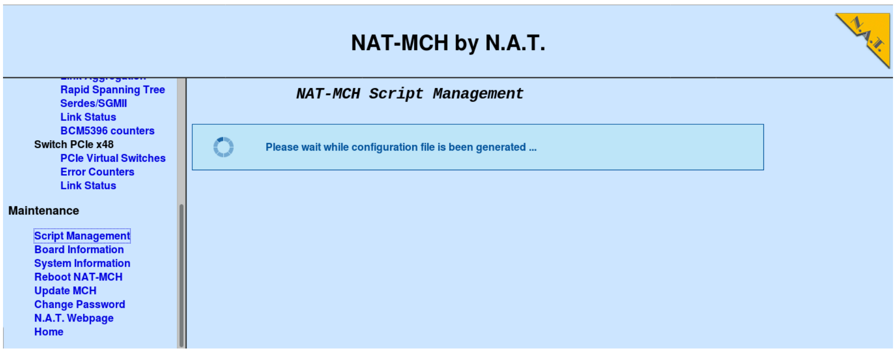
4.1 MCH configuration

Note: if you have a μTCA system set up by ICS, the Concurrent CPU and EVR are already in the correct slots and the MCH is already configured for those slots. If this is the case please jump to Section 4.2. In the case that the crate has not been set up, you will need to configure the MCH as explained in this Section.

To configure the MCH, it is necessary to access its web interface. You can read how to access it in [2].

In the menu on the left, click in `Script Management` under `Maintenance`. The MCH will take a couple of seconds to prepare the configuration script, as shown in Figure 3.

Once the script is generated, the page will look like in Figure 4. The best way of configuring the MCH is to take the running configuration and modify it according to



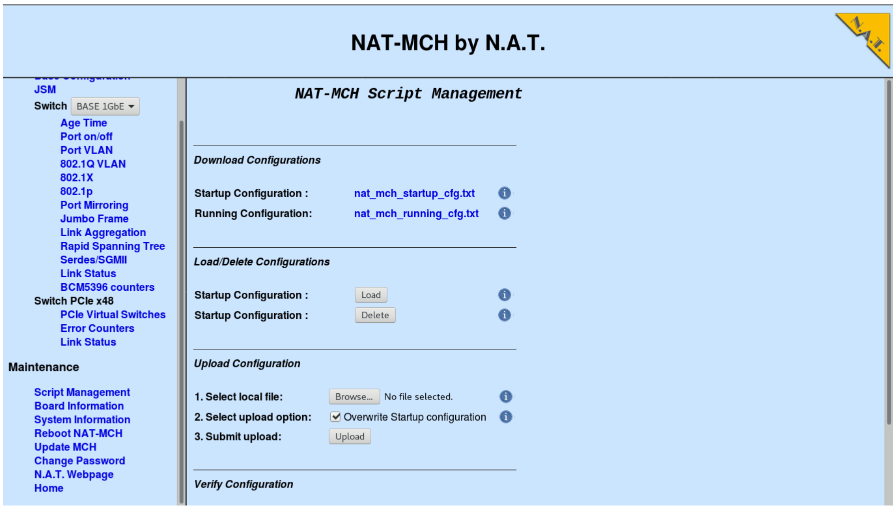
NAT-MCH by N.A.T.

NAT-MCH Script Management

Please wait while configuration file is been generated ...

Figure 3 Configuration script being generated.

our needs. To do this click on `nat_mch_running_cfg.txt` and download the file.



NAT-MCH by N.A.T.

NAT-MCH Script Management

Download Configurations

Startup Configuration : nat_mch_startup_cfg.txt

Running Configuration: nat_mch_running_cfg.txt

Load/Delete Configurations

Startup Configuration : Load

Startup Configuration : Delete

Upload Configuration

1. Select local file: Browse... No file selected.
2. Select upload option: ☑ Overwrite Startup configuration
3. Submit upload: Upload

Verify Configuration

Figure 4 Script management menu.

Open the file and look for the clock output configuration (`clk_phys_out`) section:

```
#
# Item <<clk_phys_out>>: clock output configuration
#
# Syntax: clk_phys_out = dst, src
#
# NOTE: For optimal jitter performance unused inputs should be configured as
#       outputs driving a signal with minimum signal transitions!
#
# HCSL buffer detected
# - any source for CLK 3 output other then 0 will enable output of HCSL buffer
#
# Params: dst: destination clock identifier
#                1 -   CLK1 AMC  1
#                2 -  CLK1 AMC  2
#                3 -  CLK1 AMC  3
#                4 -  CLK1 AMC  4
#                5 -  CLK1 AMC  5
#                6 -  CLK1 AMC  6
#                7 -  CLK1 AMC  7
#                8 -  CLK1 AMC  8
#                9 -  CLK1 AMC  9
#               10 -  CLK1 AMC 10
#               11 -  CLK1 AMC 11
#               12 -  CLK1 AMC 12
#                          X
#                          X
#                          X
#                          X
#               17 -  CLK2 AMC  1
#               18 -  CLK2 AMC  2
#               19 -  CLK2 AMC  3
#               20 -  CLK2 AMC  4
#               21 -  CLK2 AMC  5
#               22 -  CLK2 AMC  6
#               23 -  CLK2 AMC  7
#               24 -  CLK2 AMC  8
#               25 -  CLK2 AMC  9
#               26 -  CLK2 AMC 10
#               27 -  CLK2 AMC 11
#               28 -  CLK2 AMC 12
#                          X
#                          X
#                          X
#                          X
#               33 -  CLK3 AMC  1
#               34 -  CLK3 AMC  2
#               35 -  CLK3 AMC  3
#               36 -  CLK3 AMC  4
#               37 -  CLK3 AMC  5
#               38 -  CLK3 AMC  6
#               39 -  CLK3 AMC  7
#               40 -  CLK3 AMC  8
#               41 -  CLK3 AMC  9
#               42 -  CLK3 AMC 10
#               43 -  CLK3 AMC 11
#               44 -  CLK3 AMC 12
#               48 -  EXT single ended 2 (OUTPUT SMA 1)
#               50 -  EXT single ended 4 (OUTPUT SMA 2)
#
# Params: src: source clock identifier
#                0 -  disabled
#                1 -  CLK1 AMC  1
#                2 -  CLK1 AMC  2
#                3 -  CLK1 AMC  3
#                4 -  CLK1 AMC  4
#                5 -  CLK1 AMC  5
#                6 -  CLK1 AMC  6
#                7 -  CLK1 AMC  7
#                8 -  CLK1 AMC  8
#                9 -  CLK1 AMC  9
#               10 -  CLK1 AMC 10
#               11 -  CLK1 AMC 11
#               12 -  CLK1 AMC 12
#                          X
#                          X
#                          X
#                          X
#               17 -  CLK2 AMC  1
#               18 -  CLK2 AMC  2
#               19 -  CLK2 AMC  3
#               20 -  CLK2 AMC  4
#               21 -  CLK2 AMC  5
#               22 -  CLK2 AMC  6
#               23 -  CLK2 AMC  7
#               24 -  CLK2 AMC  8
#               25 -  CLK2 AMC  9
#               26 -  CLK2 AMC 10
#               27 -  CLK2 AMC 11
#               28 -  CLK2 AMC 12
#               35 -  EXT single ended 1 (INPUT  SMA 1)
#               37 -  EXT single ended 3 (INPUT  SMA 2)
#               41 -  100MHz OSC (only with HCSL option)
#

clk_phys_out =  1,  0
clk_phys_out =  2,  0
clk_phys_out =  3,  0
clk_phys_out =  4,  0
clk_phys_out =  5,  0
clk_phys_out =  6,  0
clk_phys_out =  7,  0
clk_phys_out =  8,  0
clk_phys_out =  9,  0
clk_phys_out = 10,  0
clk_phys_out = 11,  0
clk_phys_out = 12,  0
clk_phys_out = 13,  0
clk_phys_out = 14,  0
clk_phys_out = 15,  0
clk_phys_out = 16,  0
clk_phys_out = 17,  0
clk_phys_out = 18,  0
clk_phys_out = 19,  0
clk_phys_out = 20,  0
clk_phys_out = 21,  0
clk_phys_out = 22,  0
clk_phys_out = 23,  0
clk_phys_out = 24,  0
clk_phys_out = 25,  0
clk_phys_out = 26,  0
clk_phys_out = 27,  0
clk_phys_out = 28,  0
clk_phys_out = 29,  0
clk_phys_out = 30,  0
clk_phys_out = 31,  0
clk_phys_out = 32,  0
clk_phys_out = 33,  0
clk_phys_out = 34,  0
clk_phys_out = 35,  0
clk_phys_out = 36,  0
clk_phys_out = 37,  0
clk_phys_out = 38,  0
clk_phys_out = 39,  0
clk_phys_out = 40,  0
clk_phys_out = 41,  0
clk_phys_out = 42,  0
clk_phys_out = 43,  0
clk_phys_out = 44,  0
clk_phys_out = 48,  0
clk_phys_out = 50,  0
```

This section tells the MCH how to configure the clock lines. Although it is not mandatory, the standard is to use line TCLKB/CLK2 to send the clock from the source (the EVR in our case) to the MCH, and line TCLKA/CLK1 to distribute that same clock to the rest of the AMCs, so doing it in this way is encouraged. First check what AMC slot the EVR is installed in (slot 3 for this example, as can be seen in Figure 2), and look what identifier code is related to that AMC slot for CLK2 in the MCH's configuration script `Params: src: source clock identifier` subsection of the `clock output configuration` section (`19 - CLK2 AMC 3` in our example). Then in the same way find the identifier code of the AMC slot you want to send the event clock to in the `Params: dst: destination clock identifier` subsection (in our example AMC slots 1 to 12, except 3 where the EVR is installed).

Next do the same with `source clock identifier 100MHz OSC (only with HCSL option)` (41) and `destination clock identifier` CLK3 AMC 1 to 12 (33-44).

Finally for every destination identifier write a line with the following format:

```
clk_phys_out =  destination_clock_identifier  source_clock_identifier
```

right at the end of the clock output configuration (`clk_phys_out`) section. **Note that there is a bug in some firmware (FW) versions so the MCH will show an error when selecting the same identifier as source as destination, so that line will have to left at source 0 in that FW.**

In our example the section will look like this:

```
#
# Item <<clk_phys_out>>: clock output configuration
#
# Syntax: clk_phys_out = dst, src
#
# NOTE: For optimal jitter performance unused inputs should be configured as
#       outputs driving a signal with minimum signal transitions!
#
# HCSL buffer detected
# - any source for CLK 3 output other then 0 will enable output of HCSL buffer
#
# Params: dst: destination clock identifier
#                1 -  CLK1 AMC  1
#                2 -  CLK1 AMC  2
#                3 -  CLK1 AMC  3
#                4 -  CLK1 AMC  4
#                5 -  CLK1 AMC  5
#                6 -  CLK1 AMC  6
#                7 -  CLK1 AMC  7
#                8 -  CLK1 AMC  8
#                9 -  CLK1 AMC  9
#               10 -  CLK1 AMC 10
#               11 -  CLK1 AMC 11
#               12 -  CLK1 AMC 12
#                          X
#                          X
#                          X
#                          X
#               17 -  CLK2 AMC  1
#               18 -  CLK2 AMC  2
#               19 -  CLK2 AMC  3
#               20 -  CLK2 AMC  4
#               21 -  CLK2 AMC  5
#               22 -  CLK2 AMC  6
#               23 -  CLK2 AMC  7
#               24 -  CLK2 AMC  8
#               25 -  CLK2 AMC  9
#               26 -  CLK2 AMC 10
#               27 -  CLK2 AMC 11
#               28 -  CLK2 AMC 12
#                          X
#                          X
#                          X
#                          X
#               33 -  CLK3 AMC  1
#               34 -  CLK3 AMC  2
#               35 -  CLK3 AMC  3
#               36 -  CLK3 AMC  4
#               37 -  CLK3 AMC  5
#               38 -  CLK3 AMC  6
#               39 -  CLK3 AMC  7
#               40 -  CLK3 AMC  8
#               41 -  CLK3 AMC  9
#               42 -  CLK3 AMC 10
#               43 -  CLK3 AMC 11
#               44 -  CLK3 AMC 12
#               48 -  EXT single ended 2 (OUTPUT SMA 1)
#               50 -  EXT single ended 4 (OUTPUT SMA 2)
#
# Params: src: source clock identifier
#                0 -  disabled
#                1 -  CLK1 AMC  1
#                2 -  CLK1 AMC  2
#                3 -  CLK1 AMC  3
#                4 -  CLK1 AMC  4
#                5 -  CLK1 AMC  5
#                6 -  CLK1 AMC  6
#                7 -  CLK1 AMC  7
#                8 -  CLK1 AMC  8
#                9 -  CLK1 AMC  9
#               10 -  CLK1 AMC 10
#               11 -  CLK1 AMC 11
#               12 -  CLK1 AMC 12
#                          X
#                          X
#                          X
#                          X
#               17 -  CLK2 AMC  1
#               18 -  CLK2 AMC  2
#               19 -  CLK2 AMC  3
#               20 -  CLK2 AMC  4
#               21 -  CLK2 AMC  5
#               22 -  CLK2 AMC  6
#               23 -  CLK2 AMC  7
#               24 -  CLK2 AMC  8
#               25 -  CLK2 AMC  9
#               26 -  CLK2 AMC 10
#               27 -  CLK2 AMC 11
#               28 -  CLK2 AMC 12
#               35 -  EXT single ended 1 (INPUT  SMA 1)
#               37 -  EXT single ended 3 (INPUT  SMA 2)
#               41 -  100MHz OSC (only with HCSL option)
#

clk_phys_out =  1,  19
clk_phys_out =  2,  19
clk_phys_out =  3,  0
clk_phys_out =  4,  19
clk_phys_out =  5,  19
clk_phys_out =  6,  19
clk_phys_out =  7,  19
clk_phys_out =  8,  19
clk_phys_out =  9,  19
clk_phys_out = 10,  19
clk_phys_out = 11,  19
clk_phys_out = 12,  19
clk_phys_out = 13,  0
clk_phys_out = 14,  0
clk_phys_out = 15,  0
clk_phys_out = 16,  0
clk_phys_out = 17,  0
clk_phys_out = 18,  0
clk_phys_out = 19,  0
clk_phys_out = 20,  0
clk_phys_out = 21,  0
clk_phys_out = 22,  0
clk_phys_out = 23,  0
clk_phys_out = 24,  0
clk_phys_out = 25,  0
clk_phys_out = 26,  0
clk_phys_out = 27,  0
clk_phys_out = 28,  0
clk_phys_out = 29,  0
clk_phys_out = 30,  0
clk_phys_out = 31,  0
clk_phys_out = 32,  0
clk_phys_out = 33,  41
clk_phys_out = 34,  41
clk_phys_out = 35,  41
clk_phys_out = 36,  41
clk_phys_out = 37,  41
clk_phys_out = 38,  41
clk_phys_out = 39,  41
clk_phys_out = 40,  41
clk_phys_out = 41,  0
clk_phys_out = 42,  41
clk_phys_out = 43,  41
clk_phys_out = 44,  41
clk_phys_out = 48,  0
clk_phys_out = 50,  0
```

The script has to be uploaded to the MCH's web interface. In the `Script Management` menu, under the `Upload Configuration` section, click on `Select local file:  Browse...` and select the configuration file, check (if not already) the `Select upload option: Overwrite Startup configuration`, and click `Submit upload:Upload`.

Finally go to the `Base Configuration` on the left menu, and in the `Clock module parameter` section select `load from FLASH`. Go all the way down and click on `Save`.

After these steps the MCH should be properly configured. If there is some error try power cycling the MCH.

4.2 mTCA-EVR-300(U) configuration

For a general introduction to the EVR, please check [3]. **Note that, as shown in Table 2, the E3 version of mrfioc2 is needed.**

Start an EVR Input/Output Controller (IOC) as usual and set the following records as shown; this assumes TCLKB/CLK2 is used to send the clock from the EVR to the MCH as previously explained:

```
# Set TCLKB to low, enable it and power it up
dbpf $(IOC)-$(DEV1):OutTCLKB-Src-SP 63
dbpf $(IOC)-$(DEV1):OutTCLKB-Ena-Sel 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pwr-Sel 1

# TCLKB is 40-bit pattern, set the starting 20 bits to 1 (and the rest to 0 - default)
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.BF 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.BE 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.BD 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.BC 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.BB 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.BA 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B9 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B8 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B7 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B6 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B5 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B4 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B3 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B2 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B1 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low00_15-SP.B0 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low16_31-SP.BF 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low16_31-SP.BE 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low16_31-SP.BD 1
dbpf $(IOC)-$(DEV1):OutTCLKB-Pat-Low16_31-SP.BC 1
```

4.3 AMC configuration

If everything is set as explained before, TCLKB/CLK2 will deliver the event clock from the EVR to the MCH, and the MCH will then distribute the clock to the AMCs using TCLKA/CLK1. Configure the AMC so that it is expecting the clock in this line.

Glossary

Term	Definition
AMC	Advanced Mezzanine Card
CPU	Central Processing Unit
CU	Cooling unit
E3	ESS EPICS environment
EPICS	Experimental Physics and Industrial Control System
ESS	European Spallation Source
EVM	Event Master
EVR	Event Receiver
FW	Firmware
ICS	Integrated Control System
I/O	Input/Output
IOC	Input/Output Controller
MCH	MTCA Carrier Hub
MRF	MicroResearch Finland
MTCA	Micro Telecommunications Computing Architecture
PM	Power module
RF	Radio-frequency
SFP	Small Form-factor Pluggable

Bibliography

[1] MRF Technical Reference. *Event System with Delay Compensation Technical Reference Firmware 0205*, April 26, 2016.

[2] NAT-MCH-PHYS Technical HowTos, 2020. https://confluence.esss.lu.se/display/HAR/NAT-MCH-PHYS+Technical+HowToS#NAT-MCH-PHYSTechnicalHowToS-WebInterface (accessed 4 february 2020).

[3] Javier Cereijo Garcia and Jeong Han Lee. ESS-0508480: ICS Engineering Manual for MRF MTCA-EVR-300, 2020.

Document revision history

Revision	Reason for and description of change	Author	Date
1	First release	Javier Cereijo Garcia	February 11, 2020